Remarks

Allowance of pending claims 1-18 is respectfully requested.

35 U.S.C. §101:

The Office Action alleges a 35 U.S.C. §101 issue regarding claims 5, 14 and 16 because the "apparatus is at best a software system, *per se*, failing to be tangibly embodying or include any recited hardware as part of the apparatus." This rejection is respectfully, but most strenuously traversed for two reasons. First, the U.S. Court of Appeals for the Federal Circuit has expressly stated that computer software programs, including methods of doing business implemented as computer software programs, are patentable subject matter under U.S. Patent Law. See State Street Bank & Trust Co. v. Signature Financial Group, Inc., 1492 F.3d 1368 (July, 1998). In State Street, the Federal Circuit held that the questions of whether or not claims constitute such subject matter should not focus on the practical utility of the claimed invention, and thus, confirmed that "anything under the sun made by man" is patentable. Secondly, in this case, the apparatus claims 5, 14 and 16 at issue recite "means for" language, which is specifically authorized in paragraph 6 of 35 U.S.C. §112, which states:

> An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recited structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.

Applicants depict in FIG. 1 a computer processing environment and describe in the accompanying text that the logic presented therein is implemented within, in one example, this computer processing environment. Specific to the rejection, the means for functionality recited in independent claims 5, 14 and 16 is implemented within the depicted computer processor. As such, these claims clearly meet the requirements of 35 U.S.C. §101 and thus reconsideration and withdrawal of the rejection based thereon is requested.

35 U.S.C. §103(a):

In the Office Action, claims 1-3 and 5-7 were rejected under 35 U.S.C. §103(a) as being unpatentable over IBM Technical Disclosure Bulletin entitled "Weak Locks with Two-Level Locking Multi-Computer System Protocol to Reduce Lock-Holding Time" (hereinafter IBM TDB); while claims 4 and 8-18 were rejected under 35 U.S.C. §103(a) as being unpatentable over IBM TDB in view of Clark (U.S. Patent No. 6,598,068). Each of these rejections is respectfully traversed and reconsideration thereof is requested.

An "obviousness" determination requires an evaluation of whether the prior art taken as a whole would suggest the claimed invention taken as a whole to one of ordinary skill in the art. In evaluating claimed subject matter as a whole, the Federal Circuit has expressly mandated that functional claim language be considered in evaluating a claim relative to the prior art. Applicants respectfully submit that the application of these standards to the independent claims presented herewith leads to the conclusion that the recited subject matter would not have been obvious to one of ordinary skill in the art based on the IBM TDB and the Clark patent.

Applicants' invention is directed in one aspect (e.g., claims 1 & 5), to a method for a shared memory model system wherein a plurality of threads exists, and a bit that represents a lock type and an identifier for a thread that has acquired a lock in accordance with a first lock type, or an identifier of a second lock type, are stored in a storage area that corresponds to an object, and wherein a lock on an object is thus managed. The method includes: determining, if a second thread attempts to acquire a lock on a specific object that is held by a first thread, whether a bit that represents the lock type on a specific object represents the first lock type; setting a contention bit if the bit represents the first lock type; determining, before the first thread unlocks the specific object, whether the bit that represents the lock type represents the first lock type; storing in the storage area a special identifier that differs from the identifiers for the plurality of threads; issuing a synchronization command for the memory system; storing in the storage area data indicating the absence of a thread that holds the lock on the specific object; determining whether the contention bit has been set if the bit that represents the lock type represents the first lock type; and terminating an unlocking process if the contention bit has not been set without any other process being performed.

Initially, Applicants note that the Office Action relies in part on an Official Notice taken with respect to the subject matter of the IBM TDB and its alleged applicability to the environment of Applicants' claimed invention. This Official Notice is respectfully traversed. MPEP §2144.3 states, in part:

> Official Notice without documentary evidence to support an Examiner's conclusion is permissible only in some circumstances. ... Official Notice unsupported by documentary evidence should only be taken by the Examiner where the facts asserted to be well known, or to be common knowledge in the art, are capable of instant and unquestionable demonstration as being well-known. ... It would not be appropriate for the Examiner to take Official Notice of facts without citing a prior art reference where the facts asserted to be well known are not capable of instant and unquestionable demonstration as being well known.

The Official Notice taken in the Office Action alleges that the term "transactions" is considered functionally equivalent to Applicants' claimed "threads". This conclusion is respectfully traversed. The IBM TDB discusses a transactional environment and presents protocol for processing multiple threads using weak locks and strong locks to reduce lock holding time. The IBM TDB presents a two-phase commit lock for database transactions. In contrast, Applicants' claimed invention is to a multi-threaded processing environment and to protocol for managing a lock on a object. Applicants' claims recite functionality directed to a mutual exclusion lock. Applicants respectfully submit that the IBM TDB and the present application address locking and synchronization under completely different circumstances, and are studied by completely different communities in the art.

One skilled in the art would not look to or employ a two-phase commit lock in the transactional environment described by the IBM TDB for or in place of a mutual exclusion lock in a multi-threaded environment, nor could a mutual exclusion lock of a multi-threaded environment be used in a two-phase commit lock for a transactional environment. Since the environments are different, and the lock processes employed are different, Applicants respectfully traverse the Official Notice taken in the Office Action regarding the applicability of the IBM TDB protocol for the transactional environment as being applicable to a multi-threaded environment such as recited by Applicants.

Should the Examiner continue to entertain reservations regarding the allowability of any claim presented, the Examiner is requested to more specifically document this Official Notice pursuant to 37 C.F.R. §1.104(d)(2) and MPEP §2144.03C.

In addition to the clear difference between Applicants' claimed environment and the transactional processing environment of the IBM TDB, Applicants respectfully submit that various functionality recited in their independent claims, and in particular, in independent claims 1 & 5, is simply not taught or suggested by the IBM TDB.

For example, Applicants' recited protocol includes determining whether a bit that represents the lock type on the specific object represents the first lock type. A careful reading of the eighth paragraph of the IBM TDB (cited in the Office Action) fails to uncover any discussion of a bit representing a lock type on an object of a specific type. The IBM TDB generally states that if a NAK is returned for a K-MODE request, the transaction manager marks this, since it indicates that a contention with another transaction has occurred and the transaction will have to be re-run. There is no determination of whether a bit that represents the lock type on a specific object represents a first lock type.

Next, Applicants recite setting a contention bit if the bit represents the first lock type. The eighth paragraph of the IBM TDB is again cited in the Office Action for alleged teaching of this protocol. However, the eighth paragraph does not discuss setting a contention bit, *per se*, when another bit represents the first lock type. The IBM TDB simply teaches that the transaction manager marks when a contention exists, but not by setting a contention bit if a prior determined bit represents a first type of lock (as opposed to any lock, *per se*). These first two steps of Applicants' recited protocol are to enhance lock acquisition, while the remaining portions of claims 1 & 5 facilitate processing of lock release.

For example, the third through the eighth elements of Applicants' independent claims 1 & 5 recite issuing a synchronization command for the memory system. No similar functionality is taught by the IBM TDB. The Office Action generally references paragraph 6, pages 287-288

of the IBM TDB for such functionality, however, a careful reading of that paragraph fails to uncover any corresponding process.

For at least the above reasons, Applicants respectfully request reconsideration of the obviousness rejection to independent claims 1 & 5 based upon the teachings of the IBM TDB. Dependent claims 2-3 & 6-7 are believed patentable for the same reasons as independent claims 1 & 5, as well as for their own additional characterizations. For example, the particular functionality recited in claims 2 & 6 further characterizes Applicants' protocol for the multi-threaded environment wherein there is a thread waiting operation, a waiting thread and specific functionality that is performed in relation to that thread. The IBM TDB, to any extent deemed applicable to Applicants' environment, describes a transactional environment wherein "transaction will have to be re-run" (see paragraph 8, page 288). In the IBM TDB environment, there is no waiting thread (nor even a waiting transaction), let alone the specific protocol recited by Applicants in the claims presented. Page 289, paragraph 10 of the IBM TDB is cited in part for an alleged teaching of this functionality. However, the cited paragraph is not relevant to the protocol of Applicants' claimed invention.

With respect to claims 4 & 8-18, Applicants respectfully traverse the obviousness rejection of independent claims 9, 11, 14 & 16 for the same reasons set forth above in connection with the rejection to claims 1 & 5. The IBM TDB environment is not analogous to that of Applicants' claimed environment, and the Official Notice taken in the Office Action is traversed for the reasons set forth above. Should the Examiner continue to entertain reservation regarding the allowability of any claim presented, the Examiner is requested to produce more specific authority for the statements presented in the Office Action regarding the functional equivalency of "transactions" in a transactional environment and Applicants' claimed "thread" processing in a multi-threaded processing environment.

With respect to dependent claims 4 & 8, Applicants respectfully submit that the Clark patent does not teach any of the above-noted deficiencies of the IBM TDB when applied against independent claims 1 & 5. This patent is cited for allegedly disclosing at Col. 10, lines 5-42 Applicants' further characterization that the second lock type is a lock method whereby a queue

is employed to manage a thread that has locked access to an object. Without acquiescing to this characterization of the teachings of Clark, Applicants respectfully submit that the patent does not address or suggest the various deficiencies of the IBM TDB noted above when applied against Applicants' specific protocol recited in the independent claims. For this reason, withdrawal of the rejection of claims 4 & 8 is respectfully requested.

Claims 9 & 14 recite a technique for a shared memory model system wherein there are a plurality of threads, and a bit that represents a lock. The bit is stored in the storage area that corresponds to an object, and the system includes a queue of a thread that accesses the object. The technique includes determining, when a second thread attempts to acquire a lock on a specific object that a first thread has already locked, that a bit that is used to represent the lock on the object represents the locked state; thereafter changing data for the number of queues of threads that access the specific object in storing the updated data when the bit represents the locked state; storing the second thread in a queue and shifting the second thread to control state, for a mechanism that performs a waiting operation for accessing the specific object and a recovery operation by transmitting a notification; storing the bit that represents the locked state in the storage area before the first thread unlocks the object; determining whether a thread that is stored in a queue is present; shifting the first thread to a notification state, wherein the transmission of a notification to the thread that is waiting to be initiated, when a thread that is stored in a queue is present; and permitting the first thread to exit the notification state. Applicants respectfully submit that numerous aspects of this process are simply not taught or suggested by the applied art.

Initially, the applicability of the IBM TDB to Applicants' claimed environment is respectfully traversed for the reasons set forth above. One skilled in the art would not look to the two-phase commit lock process for the transactional environment of the IBM TDB and somehow derive therefrom locking protocol for a multi-threaded environment.

Additionally, Applicants' independent claims 9 & 14 recite, in part, storing the second thread in a queue, and shifting the second thread to a control state, for a mechanism that performs a waiting operation for accessing the specific object and a recovery operation for

transmitting a notification. In the IBM TDB, and more particularly at pages 288 – 289, paragraphs 6-8 and 10, if contention with another transaction has occurred, then the second transaction will have to be re-run. There is no queuing of transactions, let alone queuing of threads, as recited in Applicants' protocol. The IBM TDB expressly teaches otherwise by stating that the transactions will need to be re-run. Applicants' protocol further recites storing the bit that represents the locked state in the storage area before the first thread unlocks the object. No analogous processing is described in the IBM TDB.

Applicants' independent claims 9 & 14 also recite shifting of the first thread to a notification state, wherein the transmission of a notification to the thread that is waiting is initiated. Pages 287-288, paragraph 6 & 8 of the IBM TDB (cited in the Office Action) do not describe or suggest this functionality. In these paragraphs, the second transaction is re-run, so that there is no transaction in a waiting state, and there is no discussion that the first transaction enters a notification state to initiate notification to a waiting thread. Applicants' protocol then recites permitting the first thread to exit the notification state. The Office Action cites pages 288-289, paragraphs 7, 8 & 11 of the IBM TDB for alleged teaching of this functionality. Applicants respectfully submit that a careful reading of the cited paragraphs fails to uncover any discussion of a notification state previously entered by a first thread that is then exited once transmission of a notification to a waiting thread has been accomplished. There is no thread communication in the IBM TDB disclosure, nor does there appear to be any transaction communication from a first transaction to a second transaction from a notification state analogous to Applicants' recited protocol.

For all the above reasons, Applicants respectfully request reconsideration and withdrawal of the rejection to independent claims 9 & 14. Dependent claims 10 & 15 are believed allowable for the same reasons as the independent claims 9 & 14, respectively, as well as for their own additional characterizations.

Independent claims 11 & 16 again recite a protocol for a shared memory model system, wherein there are a plurality of threads, and a bit that represents a lock. The bit is stored in a storage area that corresponds to an object. A queue of threads that access the object is stored to

manage a lock on the object. The protocol includes: determining, when a second thread attempts to acquire a lock on a specific object that a first thread has locked, whether a bit that represents the lock on the object represents the locked state; changing, when the bit represents the locked state, data for the number of queues of threads that can access the specific object and storing the updated data, and thereafter issuing a synchronization command for the storage area; storing the second thread in a queue, and shifting the second thread to a control state for a mechanism that performs a waiting operation, for accessing the specific object, and a recovery operation by transmitting a notification; storing in the storage area, before the first thread unlocks the object, the bit that represents the locked state and an identifier that is not related to the representation of the locked state or an unlocked state; issuing a synchronization command for the storage area; storing, in the storage area, data that does not represent the lock on the specific object; determining whether a thread that is stored in a queue is present; shifting, when the thread that is stored in the queue is present, the first thread to a notification state wherein the transmission is initiated for issuing a notification to the thread that is waiting; and permitting the first thread to exit the notification state.

Applicants respectfully submit that the above-cited protocol of their independent claims 11 & 16 would not have been obvious to one of ordinary skill in the art based upon the IBM TDB and the Clark patent. Initially, Applicants again traverse the Official Notice taken in the Office Action equating the transactional environment of the IBM TDB to the multi-threaded processing environment recited by Applicants in the independent claims presented. As one skilled in the art, Applicants would not have looked to the transaction environment to solve the problems addressed by the present invention.

Further, numerous aspects of Applicants' protocols recited in independent claims 11 & 16 are simply not taught or suggested in the applied art. For example, Applicants recite issuing a synchronization command for the storage area after changing the data representative of the queues of threads that can access a specific object in storing the updated data. No similar functionality is described in the applied art. Further, the Office Action does not address this aspect of their recited invention.

In addition, there is no shifting, when a thread is stored in a queue, the first thread to a notification state wherein the transmission is initiated for issuing a notification to the thread that is waiting. IBM TDB, pages 287-288, paragraphs 6 & 8 (cited in the Office Action) do not teach or suggest such functionality. Again, there is no waiting transaction in the IBM TDB, but rather the second transaction must be re-run. Thus, there can be no notification from a first transaction to a second waiting transaction, *per se*. Further, there is no protocol in the IBM TDB which permits the first thread to exit the notification state. There simply is no discussion in the IBM TDB of a notification state entered by a transaction, nor is their any discussion of a first transaction initiating transmission for issuance of a notification to a second transaction in any manner analogous to the protocol recited by Applicants in their multi-threaded environment.

For all the above-noted reasons, Applicants respectfully request reconsideration and withdrawal of the obviousness rejection to independent claims 11 & 16. Dependent claims 12-13 and 17-18 are believed allowable for the same reasons as their respective independent claims, as well as for their own additional characterizations.

All claims are believed to be in condition for allowance and such action is respectfully requested.

Should the Examiner wish to discuss this case further with Applicants' attorney, the Examiner is invited to telephone their below-listed representative.

Respectfully submitted,

Kevin P. Radigan

Kevin P. Radigan, Esq.
Attorney for Applicants
Reg. No.: 31,789

Dated: September 02, 2004

HESLIN ROTHENBERG FARLEY & MESITI P.C.
5 Columbia Circle
Albany, New York 12203
Telephone: (518) 452-5600
Facsimile: (518) 452-5579